Documentation for Comparison Statistics

Documentation for ComparisonStatistics module

(a contribution Package to CDAT)

Version 3.2

PCMDI Computational Support

Program for Climate Model Diagnosis and

Intercomparison (PCMDI)

Lawrence Livermore National Laboratory

Livermore, CA 94550

United States of America

http://cdat.sf.net

10/13/2005

Table of Content

1 Introduction.

The ComparisonStatistics object calculates statistics (e.g., correlations and RMS differences) that quantify differences between two datasets. It relies heavily on the VariablesMatcher object, which is described in the VariablesMatcher documentation. Familiarity with that document is assumed here. The ComparisonStatistics package has been heavily tested under Linux (with the Portland Group Fortran Compiler), and it is believed to work under Solaris as well; it may not work on other platforms.

2 Description and Concepts.

2.1 Concepts.

The **ComparisonStatistics** object is basically a VariablesMatcher object with additional functionalities that allow the computation of basic statistics for various components and for various time_domains. As described further below, the statistics include the root−mean−square (RMS) difference and correlation between two fields, as well as means and variances of the individual fields. In this document, one field is called the "reference" field (usually an observed field), and the other the "test" field (usually a model derived field). The fields are assumed to be functions of longitude, latitude, and time. They are resolved into components including the climatology and year to year anomalies, as well as the zonal mean and deviations from the zonal mean. The analysis can be carried out for individual months, seasons, or the annual means, or statistics characterizing the "space−time" behavior can be calculated for all the months or all the seasons considered together.

2.2 The Statistics.

The following set of statistics, identified by the calling name in parentheses, is computed for 28 components:

1. test dataset mean (TestAverage)
2. reference dataset mean (ReferenceAverage)
3. test dataset variance (TestVariance)
4. reference dataset variance (ReferenceVariance)
5. correlation between test and reference datasets (Correlation)
6. root mean square difference between test and reference datasets (RootMeanSquare)
7. % of the total variance of the test dataset (TestPercentOfTotalVariance)
8. % of the total variance of the reference dataset (ReferencePercentOfTotalVariance)
9. test dataset % of the total variance of the reference dataset (TestPercentOfReferenceTotalVariance)

10. % of the total mean square difference (PercentOfTotalMeanSquareDifference)

11. rank (basically a skill score, the lower the better) = (Rank), where *rms* is the centered root−mean−square error (i.e., with the overall mean of the fields removed), and the factors in the denominator are the standard deviations of the test and reference fields.

12. weights associated with each component (Weights)

2.3 The Components.

Each field considered can be resolved into components (e.g., zonal mean, deviations from the zonal mean, etc.), and each of the above statistics can be computed for each of these components. Note, however, that because the first component is a scalar quantity, the second order statistics are not calculated in this case. Also, for some time domains (e.g., individual months or seasons), the annual cycle components (2, 4, 6, 9, 11, 13, 15, 16, 21, and 22) are identically zero.

Consider a field S^* that is a function of longitude (x), latitude (y), and time (t, with the time interval assumed to be either 1 month, 1 season, or 1 year). It is often useful to compute a seasonally varying climatology of the field (represented here by S) separately from the anomalies, S'. That is:

When treating monthly data, S^* and S' will have a time dimension equal to twelve times the number of years considered, whereas the climatology, S, will have a time dimension of length twelve.

We can further resolve the field into various spatio−temporal components, such as zonal means and annual means. To be explicit about the definition of each component, we adopt the following notation: the absence of a dimension implies that an average has been computed over that dimension, so, for example, $S(t)$ represents the spatial mean of $S(x,y,t)$ (i.e., averaged over both longitude, x, and latitude, y). Also note that "anomaly" is relative to climatology (e.g., in the case of ten years of monthly data, the January anomalies are relative to the mean of the ten January's), and t^* indicates that an annual average has been computed and for monthly data t^* increases in increments of 12 months (e.g., for 10 years of monthly data beginning with the month of March, ten annual means will be computed, with the first mean covering the period from the first March to the first February, the second covering the period from the second March to the second February, etc.) The annual means are therefore not calendar−year means, unless the first month in the time series happens to be January. Statistics for the following components are calculated by ComparisonStatistics:

#	Definition	Descriptionï¿½
1	$S( )$	Spatial and temporal mean (a scalar quantity)
2	$S(t) - S( )$	

		Climatological, spatial−mean, annual cycle (with annual mean removed)
3	*S(y) − S( )*	Climatological, annual−mean, zonal−mean (with spatial−mean removed)
4	*S(y,t) − S(y) − [S(t) − S( )]*	Climatological, annual cycle of zonal−mean (with annual−mean removed and spatial−mean annual cycle removed)
5	*S(x,y) − S(y)*	Climatological, annual−mean deviations from the zonal−mean
6	*S(x,y,t) − S(x,y) − [S(y,t) − S(y)]*	Climatological, annual cycle of deviations from the zonal−mean (with annual−mean removed)
7	*S′(x,y,t)*	Anomaly (relative to climatology)
8	*S′(t*)*	Spatial−mean, annual−mean anomaly
9	*S′(t) − S′(t*)*	Spatial−mean annual cycle anomaly (with annual−mean anomaly removed)
10	*S′(y,t*) − S′(t*)*	Annual−mean, Zonal−mean anomaly (with spatial−mean anomaly removed)
11	*S′(y,t) − S′(y,t*) − [S′(t) − S′(t*)]*	Zonal−mean, annual cycle anomaly (with annual−mean removed and spatial−mean removed)
12	*S′(x,y,t*) − S′(y,t*)*	Deviation from the zonal−mean of the annual−mean anomaly
13	*S′(x,y,t) − S′(y,t) − [S′(x,y,t*) − S′(y,t*)]*	Deviation from the zonal−mean of the annual cycle anomaly (with annual−mean removed)
14	*S(x,y)*	Climatological annual−mean
15	*S(y,t) − S(y)*	Climatological annual cycle of zonal−mean (with annual−mean removed)
16	*S(x,y,t) − S(x,y)*	Climatological annual cycle (with annual−mean removed)
17	*S(y,t)*	Climatological zonal−mean
18	*S(x,y,t) − S(y,t)*	Climatological deviations from the zonal−mean
19	*S(x,y,t)*	Climatological field
20	*S(x,y,t) + S′(x,y,t) − S( )*	"Centered" field (i.e., with time−mean, spatial−mean removed)
21	*S′(y,t) − S′(y,t*)*	Zonal−mean anomaly (with annual−mean anomaly removed)
22	*S′(x,y,t) − S′(x,y,t*)*	Deviation from the zonal−mean anomaly (with annual−mean anomaly removed)
23	*S′(y,t*)*	Zonal−mean, annual−mean anomaly
24	*S′(x,y,t*)*	Annual−mean anomaly
25	*S′(y,t)*	Zonal−mean anomaly
26	*S′(x,y,t) − S′(y,t)*	Deviation from the zonal−mean anomaly
27	*S′(x,y,t)*	Anomaly (relative to climatology)
28	*S(x,y,t) + S′(x,y,t) − S( )*	"Centered" field (i.e., with time−mean, spatial−mean removed)

ï¿½ Note that these descriptions apply to the most general case: for fields that are functions of longitude,

latitude, and time, and the time sampling frequency is monthly or seasonally (i.e., 12 months per year or 4 seasons per year). If individual months or seasons are considered (e.g., a series of Januaries or a series of summers), then there is no distinction between t and t^* (e.g., $S'(t) = S'(t^*)$) and the climatological components are time−independent (e.g., $S(t) = S( )$). In this case the annual cycle components (2, 4, 6, 9, 11, 13, 15, 16, 21, and 22) are identically zero, and the descriptive phrase "annual−mean" is not appropriate, and should be replaced simply by "mean" in the case of climatological fields and omitted entirely in the case of anomaly fields.

Note that only 12 of these components are independent (i.e., mutually orthogonal): 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, and 13); the others are linear combinations of these. Also note that components 7 and 27 are identical and components 20 and 28 are identical (the redundancy originally served to simplify writing of a table).

2.4 The Time Domains.

When computing the statistics, various time domains can be selected. Note that when a single month or season or the annual mean case is selected, the climatological statistics are purely spatial statistics. Otherwise the statistics are space−time statistics, in general calculated over longitude, latitude and time.

1. January
2. February
3. March
4. April
5. May
6. June
7. July
8. August
9. September
10. October
11. November
12. December
13. DJF

14. MAM

15. JJA

16. SON

17. Annual

18. Seasonal Space−Time

19. Monthly Space−Time

3 Usage.

3.1 Construction.

Construction is identical to a Compare object:

c=ComparisonStatistics(ReferenceVariableConditioner, TestVariableConditioner, [optional arguments])

3.2 Time specific options

Two parameters (minyr and fracmin) determine whether the number of time−samples available from a grid cell is large enough to include that cell in the analysis:

minyr = the minimum number of years of data needed for the grid cell to be included in the calculation of the statistics, but this condition is annulled if fracmin is exceeded. The default value of **minyr** is 3.

fracmin = the minimum fraction of years of data required for the grid cell to be included in the calculation, but this condition is annulled if minyr is exceeded. The fraction of years is the number of years of data available (i.e., not missing) divided by the number of years in the time−domain considered. The default value of fracmin is 0.5 .

For example to reset these options to their default values:

c.minyr=3.

c.fracmin=0.5

3.3 Computing.

Once created, computations are made by calling the compute function:

ref, test=c.compute()

You can also pass 2 (optional) arguments to the compute function:

ref, test=c.compute(**time_domain**=range(1,20), **returnTuple**=1)

time_domain is a list of values (or a single value) of the time_domains you're interested in (which by default is all domains).

test and **ref** are the 2 variable processed as they would be by the VariablesMatcher (e.g., mapped to a common grid, masked, etc.). These fields are required if you plan to invoke the write function with the ascii option (see below) or if you want to do other calculations on the fields besides those performed by ComparisonStatistics.

If **returnTuple** is set to 1, then **test** and **ref** are returned with the associated weights for each grid cell (see VariablesMatcher documentation).

3.4 Accessing the Statistics.

At this stage any statistic can be obtained by calling the function carrying its name. To obtain, for example, the variance of the test dataset:

var=c.TestVariance()

To get a specific time domain (assuming it was in the list you passed to the compute function):

var=c.TestVariance(time_domain=[4,5,6])

or

var=c.TestVariance(time_domain=5)

Also you could get simply one or a few components:

var=c.TestVariance(components=[2,3,4])

Both the time domain and components can be simultaneously specified:

var=c.TestVariance(components=[2,3,4],time_domain=[4,5,6])

The resulting variable is two dimensional and a function of the time_domain and the component.

3.5 Writing the results to a file.

The ComparisonStatistics object comes with an output option. By default, output is written in netCDF format, but if the filename ends with one of the following extensions: '.doc', '.out', '.text', '.txt', '.asc', '.asci', or '.ascii', then ASCII output is triggered. This option is included for backward compatibility with other software, but is not recommended.

3.5.a Usage.

c.write(file, comments='None', test=None, ref=None, mode='w')

where

file: output filename

comments: Additional comments you would like to add to the file (perhaps describing regridding and masking operations that were performed before the statistics were calculated).

test and ref: fields returned from c.compute() (***required*** *for ASCII output*).

mode: mode to open the netCDF file: ('w' , 'r+', 'a')

It is also possible to write only a single statistic to a netCDF file, as shown in the following example for the "test variance" statistic: Also each statistic has its own write function (for netCDF only):

c.TestVariance.write('my_netcdf_file.nc', mode='w')

3.5.b What is saved?

3.5.b.i File attributes.

The following global attributes will be included in the netCDF output files:

ï¿½ comments: (if passed by the user)

ï¿½ test_dataset: description of the test dataset (file, var, id, grids , masks, etc...)

ï¿½ reference_dataset: description of the reference dataset

ï¿½ external_dataset: description of the external dataset

ï¿½ final_grid: description of the final grid

ï¿½ time_domain: string that allows one to reconstruct the python dictionary of the time_domain names using the eval function.

ï¿½ components: string that allows one to reconstruct the python dictionary of the component names using the eval function.

3.5.b.ii Statistic Variable.

The following variables will be saved:

ï¿½ For each statistic (e.g., Correlation, Rank, Weights, etc.), an array with the shape: (component, time_domain)

ï¿½ The component dimension with integer values corresponding to those given in section 2.3.

ï¿½ The time_domain dimension with integer values corresponding to those given in section 2.4.

To extract a statistic from an opened cdms file, f, use the following method:

stat = f(statistic_name, component=component_index, time_domain=time_index)

The following, for example, will extract the correlation statistic for interannual variations (component 7) for the monthly space−time (time_domain 19):

correl=f('Correlation', component=7, time_domain=19)

4 Example.

4.1 Simple Example.

```
import ComparisonStatistics
import cdutil

# Reference
ref='/pcmdi/obs/mo/tas/jones_amip/tas.jones_amip.ctl'
Ref=cdutil.VariableConditioner(ref)
Ref.var='tas'
Ref.id='jones-ncep'

# Test
tst='/pcmdi/obs/mo/tas/rnl_ncep/tas.rnl_ncep.ctl'
Tst=cdutil.VariableConditioner(tst)
Tst.var='tas'
Tst.id='ncep'

# Final Grid
FG=cdutil.MaskedGridMaker()
FG.longitude.n=36
FG.longitude.first=0.
FG.longitude.delta=10.
FG.latitude.n=18
FG.latitude.first=-85.
FG.latitude.delta=10.

# Now the ComparisonStatistics thing
c=ComparisonStatistics.ComparisonStatistics(Ref, Tst, maskedGridMaker=FG)
c.fracmin=.5
c.minyr=3
icall=19

# Let's use indices to extract common times
# (in cases when the time model is inconsistent with CDMS)
c.VariableConditioner2.cdmsKeywords['time']=slice(252,372)
c.VariableConditioner1. cdmsKeywords['time']=slice(0,120)
print "Before computing:"
print c.VariableConditioner1
(ref,reffrc), (test,tfr) = c.compute()
print "Test:",test

# Retrieve all 28 components of the "rank" statistic
# for the time_domain 19 (monthly space time)
rank=c.rank(time_domain=19)
print 'Result for Rank:',rank
c.write('tmp.nc',comments='A simple example')
```